AD-A262 444

RL-TR-92-303
In-House Report
November 1992

# NEURAL WAVEFORM RECOGNITION SYSTEM

20001101203

Scott S. Shyne



*APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.*

93-06684 3688



Rome Laboratory
Air Force Materiel Command
Griffiss Air Force Base, New York

93 3 31 123

This report has been reviewed by the Rome Laboratory Public Affairs Office (PA) and is releasable to the National Technical Information Service (NTIS). At NTIS it will be releasable to the general public, including foreign nations.

RL-TR-92-303 has been reviewed and is approved for publication.

APPROVED: Anthony S. Szalkowski

ANTHONY S. SZALKOWSKI, Acting Chief
Communications Transmission Division
Command, Control & Communications Directorate

FOR THE COMMANDER: John A. Graniero

JOHN A. GRANIERO
Chief Scientist
Command, Control & Communications Directorate

**If your address has changed or if you wish to be removed from the Rome Laboratory mailing list, or if the addressee is no longer employed by your organization, please notify RL (C3BB) Griffiss AFB NY 13441-5700. This will assist us in maintaining a current mailing list.**

**Do not return copies of this report unless contractual obligations or notices on a specific document require that it be returned.**

# REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188



1. AGENCY USE ONLY (Leave Blank)

2. REPORT DATE
November 1992

3. REPORT TYPE AND DATES COVERED
In-House May 92 - Aug 92

4. TITLE AND SUBTITLE
NEURAL WAVEFORM RECOGNITION SYSTEM

5. FUNDING NUMBERS
PE - 62702F
PR - 4519
TA - 42
WU - 76

6. AUTHOR(S)
Scott S. Shyne

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Rome Laboratory (C3BB)
525 Brooks Road
Griffiss AFB NY 13441-4505

8. PERFORMING ORGANIZATION REPORT NUMBER
RL-TR-92-303

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
Rome Laboratory (C3BB)
525 Brooks Road
Griffiss AFB NY 13441-4505

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES
Rome Laboratory Project Engineer: Scott S. Shyne/C3BB (315) 330-4819

12a. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited.

12b. DISTRIBUTION CODE

13. ABSTRACT (Maximum 200 words)

The Communications Technology Branch in the Command, Control and Communications Directorate of Rome Laboratory performs theoretical research in the area of advanced signal processing as it applies to military communications systems. The research documented in this technical report examines the applicability of neural networks to the communications problem of waveform recognition. A basic theoretical architecture for a neural network based communications system is presented along with detailed simulation results.

14. SUBJECT TERMS
Neural Networks, Kohonen, Signal Processing

15. NUMBER OF PAGES
40

16. PRICE CODE

| 17. SECURITY CLASSIFICATION OF REPORT | 18. SECURITY CLASSIFICATION OF THIS PAGE | 19. SECURITY CLASSIFICATION OF ABSTRACT | 20. LIMITATION OF ABSTRACT |
|---|---|---|---|
| UNCLASSIFIED | UNCLASSIFIED | UNCLASSIFIED | U/L |

# Table of Contents

# List of Figures

# List of Tables

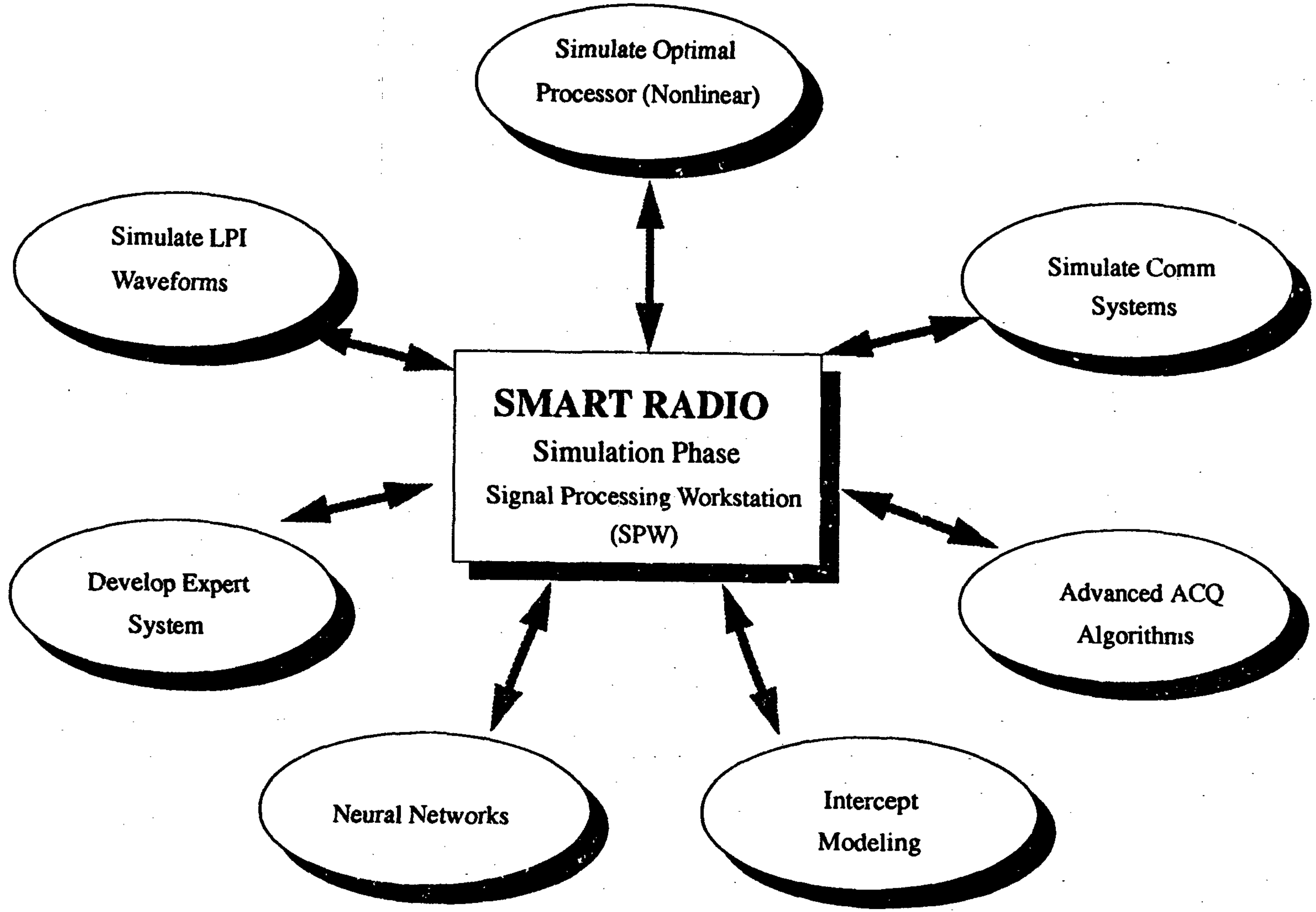
## 1.0 Introduction:

The Communications Technology Branch of Rome Laboratory performs theoretical research in the area of advanced signal processing concepts as they apply to military communications systems. Research performed by the laboratory manifests itself in several different forms. Contractual work with universities and small businesses generates important theoretical information that augments in-house research performed by Government scientists. The in-house research stimulates independent thought processes allowing scientists and engineers to derive creative applications for state-of-the-art technology. The research documented in this technical report is a conglomeration of in-house work performed by Scott Shyne (Rome Lab computer scientist) and technical data gathered from many scientific technical reports created through independent contractual efforts. The work documented in this technical report was performed under the Rome Lab In-House program "AJ/LPI Workstation", PR C-1-H403. The objective of the AJ/LPI Workstation effort is to develp, test, and evaluate advanced communications and signal processing algorithms for use in the Air Force's next generation SMART MULTIBAND radio systems (see Figure 1). A wide range of technologies are being investigated including: Low Probability of Intercept (LPI) waveform design, advanced interference cancellation techniques, advanced encoding/modulation schemes, applications of Expert Systems, and applications of neural networks.

Neural Networks are currently being investigated for various applications to the communications problem. Areas of investigation include, but are not limited to: waveform modulation/demodulation, adaptive equalization, environmental monitoring, and wavefrom recognition. This technical report is intended to illustrate the possible applicability of neural networks to the specific communications problem of waveform recognition. All referenced materials are listed in the bibliography.

## 1.1 Background:

Communications links may be completely disrupted by channel effects or through electronic jamming. Enemy detection units will attempt to intercept signals using various techniques




**Smart Radio Effort**
**Figure 1**

to exploit and decipher the signal. The capability to effectively utilize a communications circuit by selecting several different modulation types may be of paramount importance to the success of the mission. In some instances, it may be desirable for the channel to be reconfigured as a much lower bandwidth channel in order to gain an advantage against noise and to use the low bandwidth mode to communicate new coordination information such as a change frequency order.

In circumstances such as these, the receiver may not be aware that a change in modulation has occurred. A receiver, in such a system, requires the capability for real-time waveform recognition whereby a change in the channel from one modulation technique to another can be realized without severe connectivity loss. Neural paradigms excel in the recognition of patterns even in the presence of channel noise. It is theoretically possible to design a receiver that operates in a varying modulation mode by utilizing intelligent waveform recognition techniques.

This new receiver would be trained on a specified set of waveforms, dependent on the communications environment of the particular mission. The receiver would be able to change waveforms or modulation types in "real-time" without apriori knowledge of a reconfiguration event from the transmitter. The capability to change modulation techniques, whenever the situation warrants, would create a much more robust and adaptable smart radio configuration.

## 1.2 Scope:

The information presented in this paper outlines a possible architecture for a Neural Waveform Recognition System (NWRS). In-house research provided the vehicle to explore the applicability of a neural based waveform recognition module to be incorporated into an adaptable smart radio configuration. The motivation for this effort was to provide a "proof-of-concept" architecture that utilized artificial neural systems to perform signal analysis for the purpose of identifying specific modulation types. A simulation program was written in the C programming language to help determine the appropriate neural paradigm for insertion into the NWRS. Future development will allow the C code from this simulation to be linked into ComDisco's Signal Processing Workstation (SPW) simulation environment so that a full system can be designed and implemented utilizing this new module. The NWRS exploits the keen pattern recognition capabilities of neural

networks in an attempt to provide an essential module for an adaptable communications system to be utilized in a tactical radio of the 21st Century.

## 1.3 Outline of Report:

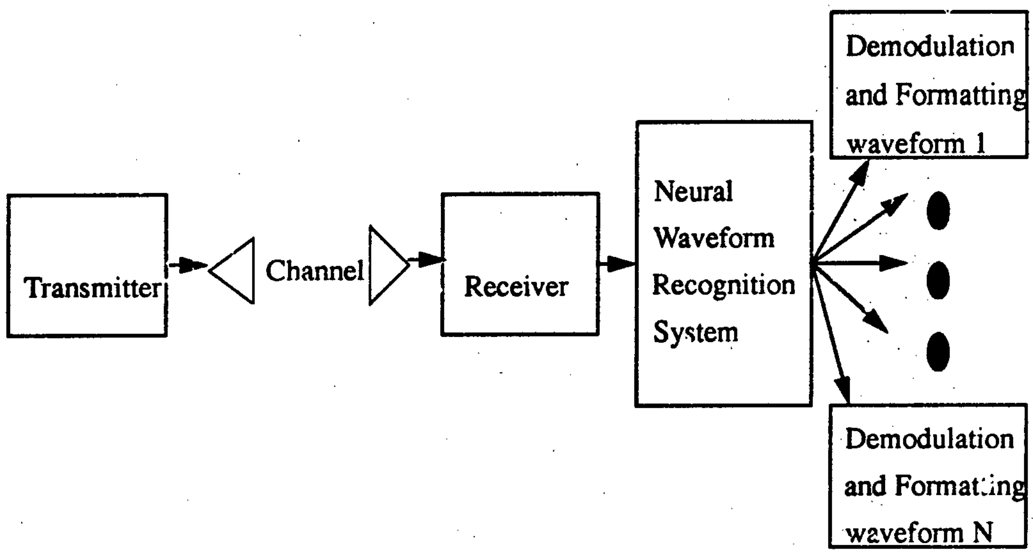
Section 1.0 gives a general overview of the background and scope of this technical report. Section 2.0 discusses an adaptable smart radio configuration designed to utilize the NWRS. Section 3.0 presents a hierarchical design of the Neural Waveform Recognition System (NWRS) including some of the motivational factors behind the design concept. Section 4.0 presents the technical details of the neural network paradigm that is utilized by the NWRS and a brief discussion of how the paradigm performs the basic signal classification task. Section 5.0 gives a detailed explanation of the simulation that was written to test the neural paradigm. This section will include a brief description of the simulation environment, compiler, processor, data files, and execution of the program. This section also discusses some of the performance analysis results from the simulation. Section 6.0 presents some of the basic results of this effort and makes recommendations for future work. The bibliography lists all the supporting documentation that was utilized throughout this research effort. The source code for the NWRS is available for perusal by contacting the author of this technical report.

## 2.0 NWRS Radio Configuration:

The NWRS Radio Configuration, shown in Figure 2, illustrates a functional block diagram of a digital communications system utilizing the Neural Waveform Recognition System (NWRS) to correctly identify the type of modulation technique inherent to the received waveform.

The basic configuration of this architecture includes the transmission, reception, recognition and demodulation/reverse formatting of baseband signals. The transmitter would transform textual information into binary digits through the use of some type of coding algorithm. Analog information is formatted using three separate processes: sampling, quantization, and coding. In all cases, the formatting step results in a sequence of binary digits. These digits are modulated into compatible baseband channel waveforms to produce a sequence of signal pulses with characteris-

tics that correspond to the binary digits being sent. The receiver detects the pulses coming across the channel and buffers the pulses into a 1024 element signal epoch. This signal epoch is then




NWRS Radio Configuration

Figure 2

passed directly into the Neural Waveform Recognition System. The NWRS attempts to classify the signal epoch into specific waveform modulation types. Once the correct waveform modulation type has been identified, the signal epoch is forwarded to the proper demodulation and reverse formatting routines. The demodulation routine produces an estimate of the transmitted digits and the reverse formatting recovers an estimate of the source information.

## 3.0 Neural Waveform Recognition System:

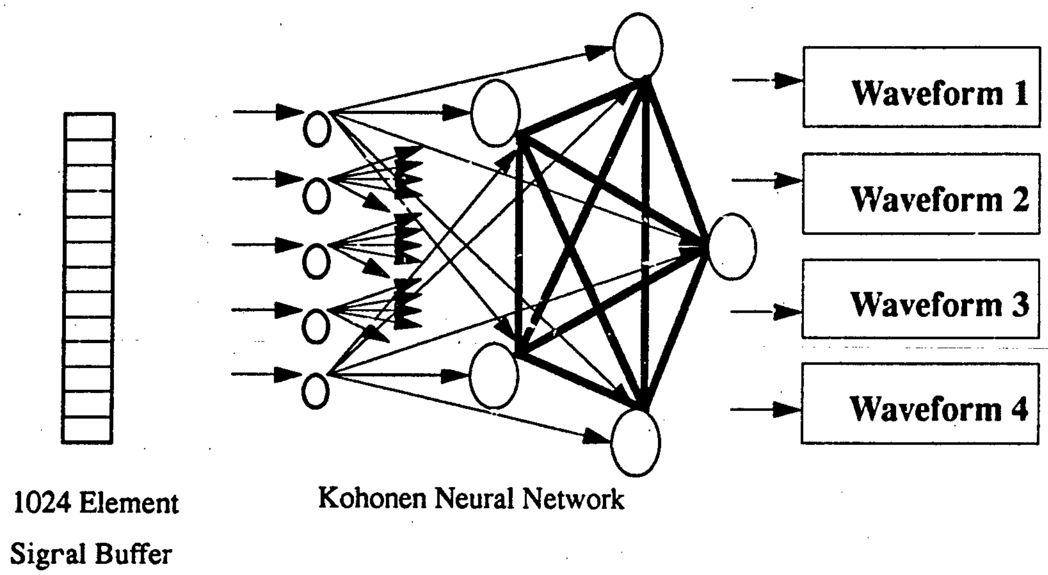
The Neural Waveform Recognition System (NWRS) is an intelligent signal recognition module that can be incorporated into an advanced tactical radio configuration. It will provide a highly advanced classification mechanism that will determine what type of modulation scheme a receiver should utilize.

## 3.1 Design Motivation:

The NWRS provides intelligent signal classification of signal pulses received from the baseband channel. This classification information results in a radio system that can switch between several various types of modulation schemes in real-time without prior notification of a changing modulation type. This capability allows the NWRS to recognize signals from various radio configurations utilizing different modulation types. By varying the type of modulation used during the transmission of a signal, the overall Low Probability of Intercept (LPI) characteristics of the signal are increased. By the time the unfriendly listener figures out the modulation type being utilized, the radio has changed modulation types several times.

## 3.2 Theoretical Architecture:

NWRS incorporates the Kohonen Self-Organizing Feature Map neural paradigm to perform the classification of the signal pulses. The network will classify a signal vector of 1024 pulses that was received by the signal buffer at the front end of the NWRS (see Figure 3).




Neural Waveform Recognition System

Figure 3

This signal vector serves as the 1024 element input vector to the neural network. The output of the network will be the proper classification of the waveform. With the correctly classified waveform, an advanced tactical radio can utilize the proper techniques to decipher the transmitted message. The eventual application might look something like Figure 3.

## 4.0 Kohonen's Self-Organizing Feature Map:

The Kohonen Self-Organizing Feature Map is a neural network paradigm that performs the k-means clustering algorithm on an n-dimensional continuous valued vector. The network goes through a period of learning where many different input vectors are presented. The network continues to train on the input vectors until it is determined that the network has successfully learned the vectors. The point at which this occurs varies widely from application to application due to the degree of granularity required for proper classification of the input patterns and also due to the orthogonality of the input patterns. Orthogonal training patterns will cause the network to learn very quickly whereas similar input patterns will require a greater amount of training time due to the fact that the network must work harder to differentiate between two similar patterns. Through repeated training iterations, the neural network forms a feature map based on the training vectors. Each training vector maps to a specified exemplar in the output layer of the network. Once each training pattern has converged to one specific exemplar, the network has successfully learned the training vector and is ready to be tested with corrupted signal vectors.

## 4.1 Background Information:

The Kohonen neural network was developed by Teuvo Kohonen of the Helsinki Technology University in Finland. In 1981, Teuvo Kohonen succeeded in defining a process which very effectively forms various abstract "topographic maps" of sensory experiences. These memory maps tend to self-organize in a similar manner to that of theoretical biological systems. The Kohonen learning process adopts its physical architecture from a model of the biological innerworkings of the human brain. The activity of every neural cell in the central nervous system depends on signals received from a great many other cells. There also exists mutual interconnections between remotely located cell groups made through long axons. The axon is that part of a

nerve cell through which impulses travel away from the cell body. These mutual interconnections are simulated in the learning algorithm by creating lateral inhibition about the output nodes. This means that nodes have the ability to "excite" or "activate" other nodes that are next to them. Nodes that are in a relative proximity of each other are referred to as belonging to the same neighborhood. By activating nodes in the same area of memory, the memory begins to self-organize by adjusting the interconnection weights, effectively partitioning the memory based on the number of patterns presented. Once the network has been trained on a given number of input patterns, no new patterns may be introduced. The original patterns can be correctly recognized by being classified in the appropriate partition of memory. Kohonen's Self-Organizing Feature Map also has the ability to recognize the patterns after being degraded by some amount of noise.

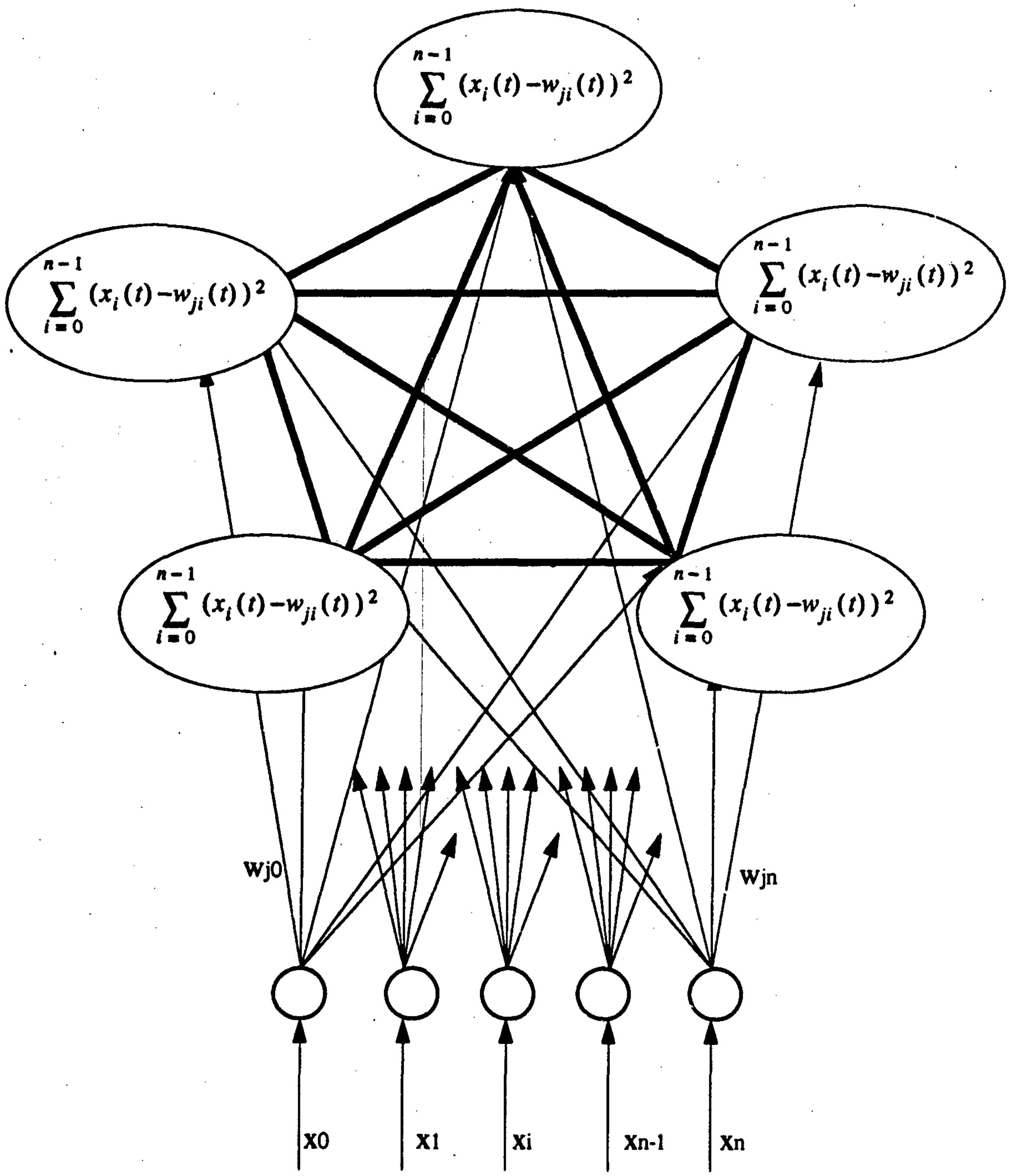
## 4.2 Technical Description:

The following two sections describe the Kohonen Self-Organizing Feature Map from a theoretical and a mathematical viewpoint. Section 4.2.1 will give the reader a conceptual understanding of how the algorithm works and Section 4.2.2 will back up the theory with the specific mathematical equations used to implement the network.

### 4.2.1 Theoretical Viewpoint:

Kohonen's Self-Organizing Feature Map contains both an input layer and an output layer (see Figure 4). In the simulation to be discussed later, there are 1024 input nodes and 1024 output nodes. The 1024 output nodes allow 1024 possible patterns to be stored. The interconnections between layers are represented by real valued connection weights. The output nodes exhibit strong lateral inhibition which means that there is a large number of virtual interconnections between the output nodes. The connections do not exist as weights but as relationships to each other. A minimum error classifier determines the best output node for a given input vector by choosing the exemplar that is the shortest distance from the input vector. Lateral inhibition (the effect a node has on other nodes in the same layer) is also exhibited by the output layer through the determination of neighborhoods for specific nodes. These neighborhoods decrease with time and as a result, so does the degree of lateral inhibition. The input layer takes a vector composed of

# Kohonen's Self Organizing Feature Map




Kohonen Architecture

Figure 4

continuous or binary data. This vector represents a pattern that could be a visual pattern or an electrical signal vector. The vector is matched to the best exemplar by determining the minimum distance from the input vector to all the exemplars. The output of the network is the best matching exemplar. By training the network on a known set of data, certain exemplars have been defined to represent a match to a specific pattern. When an exemplar has been chosen as the best match for a particular input vector, that input vector has been identified as the pattern that matched to that exemplar during the initial training.

### 4.2.2 Mathematical Viewpoint:

The Kohonen Self-Organizing Feature Map paradigm utilizes the k-means clustering algorithm to perform the classification of the input data. Six separate procedures implement the algorithm that drives the Kohonen neural network. Each of these procedures is based on simple, well-understood statistical analysis techniques.

The first procedure initializes the connection weights between the input and output layers of the network. Theses connections should be initialized to small random values (possibly ranging from 0.0 to 0.3). This initialization procedure also determines the size of the neighborhood of nodes (degree of lateral inhibition, see Para 4.2.1). At the start of training, all the output nodes are in the same neighborhood.

The second procedure presents the input vector to the network. The input vector should have exactly the same number of entries as there are input nodes. The data represents time varied sample signal points that were stored in the signal buffer. The data is analog and it defines the sample waveform over a given time period.

The third procedure computes the Euclidean distance from the input vector to each output node. This distance computation can be accomplishes in parallel, allowing for rapid computation of multiple distances. The result of each distance compilation is stored in a "distance" array. The array is given to the fourth procedure for analysis. The distance to each node is computed by the following formula:

$$d_j = \sum_{i=0}^{n-1} (x_i(t) - w_{ji}(t))^2$$

$d_j$ = all the distances from the input vector to each exemplar.

$x_i(t)$ = each entry in the input vector at time t.

$w_{ji}(t)$ = weighted connection between input node i and output node j at time t.

n - 1 = the number of input nodes - 1.

The fourth procedure of the algorithm involves selecting the minimum distance from the array generated in procedure 3. Each index in the "distance" array represents the distance from the input vector to that particular exemplar. The exemplar with the minimum distance from the input vector becomes the best matching exemplar. The following formula illustrates the minimum distance equation.

$$d_j^{\wp} = \binom{min}{j} \{d_j\}$$

$d_j^{\wp}$ = Best matching exemplar in distance array (minimum distance)

$\binom{min}{j}$ = Function used to choose the minimum distance from the input vector to the exemplar j.

$\{d_j\}$ = the array of distances from the input vector to exemplar j

The fifth procedure is the most complex algorithm component. It involves the adaptation of the interconnection weights between the input and output layers. The interconnection weights retain the overall knowledge of the input vectors. These weights are adapted based on several changing variables. The adaptation must be accomplished with precision so that the information maintained by the interconnection weights is not lost. The size of the neighborhood, as well as the amount of change to a given weight, decreases with respect to time. This gradual change is required to gracefully encapsulate the input vector knowledge into the connection weights without drastically altering the weights. By slowly changing the weights, previous knowledge is preserved. The following formula reflects the adaptive update equation for the interconnection weights of the Kohonen Self-Organizing Feature Map.

$$w_{ji}(t+1) = w_{ji}(t) + g(t)\,(x_i(t) - w_{ji}(t))$$

For all output nodes in the neighborhood of node j, where:

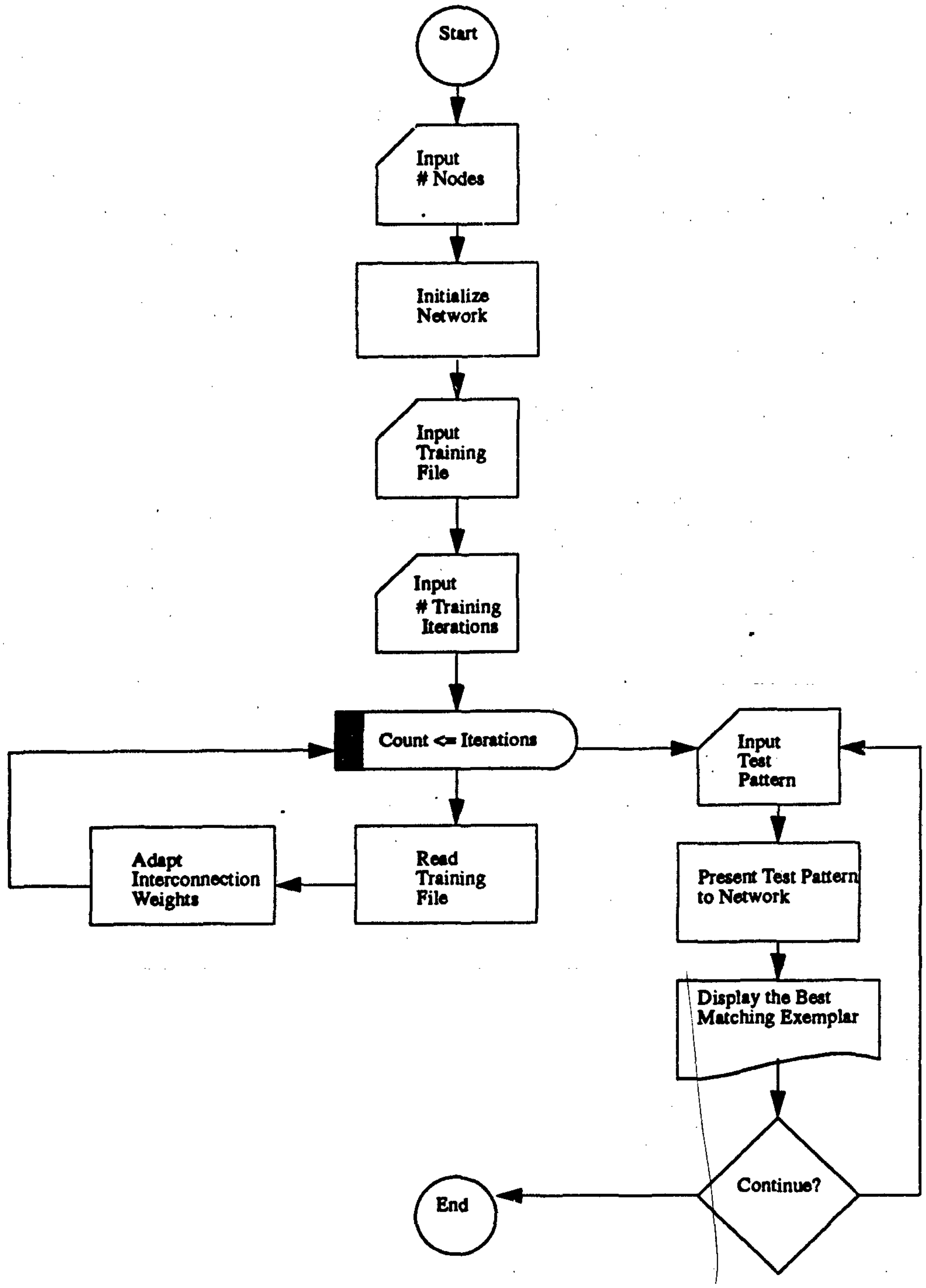
$0 \le i \le n-1$

| | |
|---|---|
| $w_{ji}(t+1)$ | New interconnection weight at time t + 1. |
| $w_{ji}(t)$ | Current interconnection weight from node i to node j at time t. |
| $g(t)$ | Gain term at time t where gain is defined as $0 < g(t) < 1$. |
| $x_i(t)$ | Ith entry in the input vector at time t. |

The sixth procedure of the algorithm is to go back to the second procedure and present the next input vector. The network continues to adapt itself and readjust the connection weights until the feature map is completely defined based on the input vectors. The feature map becomes completely defined when input vectors repeatedly match to specific exemplars in the output layer. Once training has been completed, the degree of flexibility and noise tolerance exhibited by the system can be determined by utilizing corrupted input vectors and presenting them as input to the network. The network must be able to recognize the correct waveform from the corrupted data.

## 5.0 Simulation and Analysis:

The main program is "koh1.c" and can be executed by typing koh <cr>. The execution flow is shown in Figure 5. The program asks for the number of nodes in the input layer. It automatically assumes an equal number of input nodes and output nodes. In this implementation, the input layer and the output layer contain 1024 nodes each. The 1024 output nodes allow a maximum of 1024 possible patterns to be stored. Realistically, this network would not be very efficient if more than 300 patterns were introduced. The more patterns that are stored, the more susceptable the network becomes to outside interference such as jamming or Gaussian noise. The interconnections between layers are represented by real valued connection weights. The program asks for the name of the training file. The training file contains the four waveforms. Each waveform is represented as a 1024 point vector of real numbers. The program asks for the number of training iterations. NWRS reads the input data from the training file and begins to self-configure itself utilizing




**Flow Diagram for NWRS Execution**
**Figure 5**

the Kohonen Self-Organizing Feature Map neural paradigm. Each waveform will gradually converge on a specific exemplar in the output layer feature map. This is accomplished through the use of a minimum error classifier that determines the best output node for a given input vector by choosing the exemplar that is the shortest distance from the input vector. The output of the network is the best matching exemplar. By training the network on a known set of data, certain exemplars have been defined to represent a match to a specific pattern. When an exemplar has been chosen as the best match for a particular input vector, that input vector has been identified as the pattern that matched to that exemplar during initial training. During training, lateral inhibition is exhibited by the output layer through the determination of neighborhoods for spec ^c nodes. These neighborhoods decrease with time and as a result so does the degree of lateral inhibition. These neighborhoods gradually build the feature map characteristics that provide a "near hit" capability for testing waveform recognition in the presence of noise. EXAMPLE: If a specific waveform has converged after training to centroid number 43, the correct classification of this pattern in 10 percent noise could be a centroid match of 41,42, 43, 44 or 45. Once training is complete, the adaptation of the interconnection weights is discontinued. This allows the network to retain the patterns it was initially trained on without being corrupted by noisy input data. A more detailed discussion of the testing of the NWRS is discussed in Section 5.3, Simulation Description.

## 5.1 Development Environment:

The NWRS program was written in the C programming language utilizing the C compiler available with SUNOS 4.1.1. The program was hosted on a SUN Sparcstation 2 which is based on the Reduced Instruction Set (RISC) architecture. The code has been ported to an IBM PC and an Amiga 2000 platform. The most updated version of the NWRS resides on the Sun workstation. Future implementations will involve creating a custom coded block in ComDisco's Signal Processing Workstation (SPW) allowing the simulation to be tightly coupled with other elements of communications systems enabling a more in-depth study of the NWRS's effectiveness in classifying various signals.

## 5.2 Data Generation:

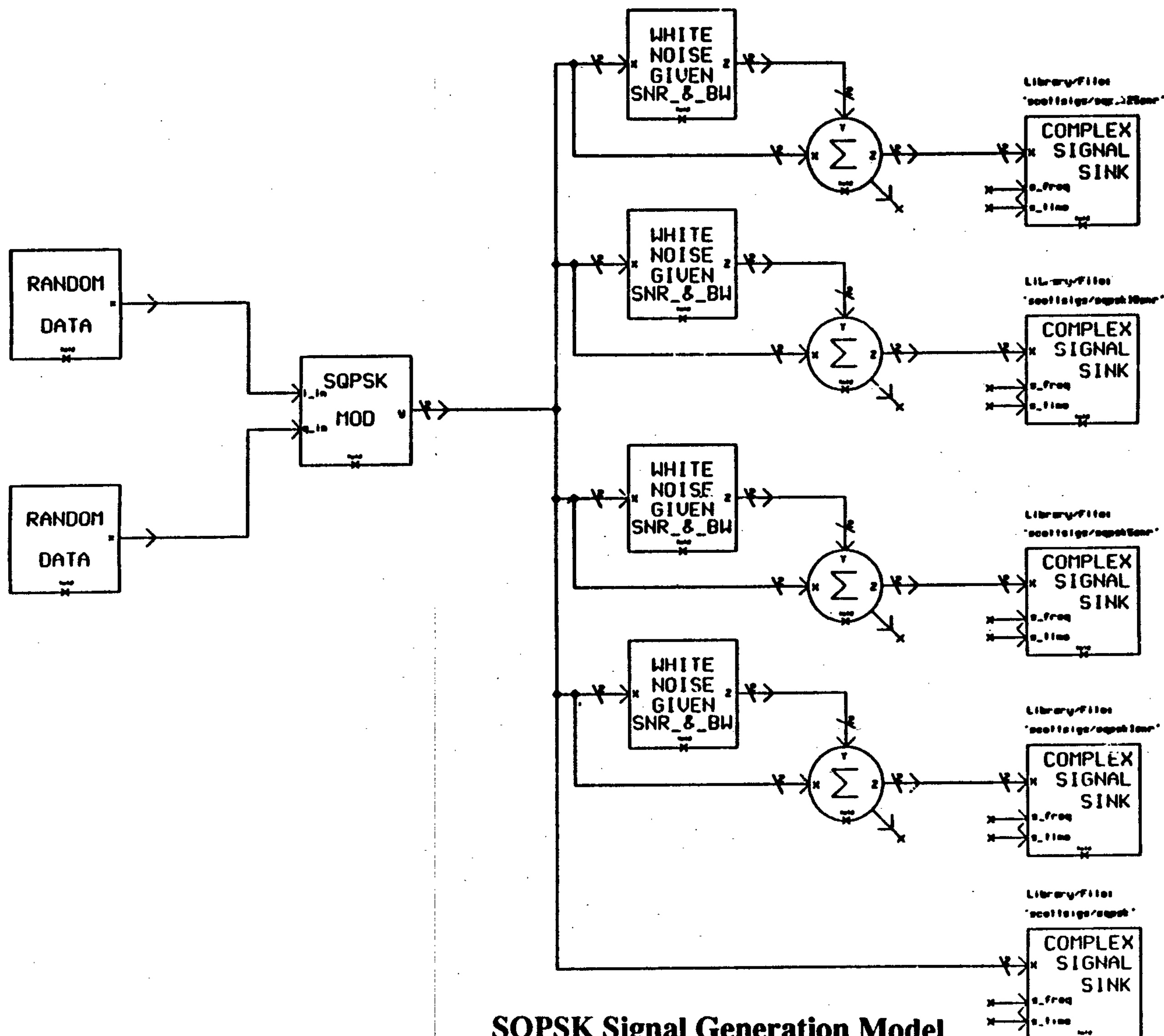
The NWRS program requires several data files for operation. These data files were initially generated by a small C program written for the express purpose of generating signal epochs for the NWRS program. The program generated a sine wave, a cosine wave and two mixtures of both the sine and cosine waveforms. The data was appropriate for initial testing and debugging of the NWRS but inappropriate for credible performance analysis during actual operation. It was decided to generate the test data by utilizing the extensive signal generation capabilities of Com-Disco's Signal Processing Workstation (SPW). Four systems were designed using SPW's Block Diagram Editor. The signals generated by each of these systems were displayed using SPW's Signal Display Editor. Each of the signals generated were written to the hard disk in ASCII format and the ASCII files were used as input to the NWRS. The four systems that were designed using SPW were based on the following modulation schemes: Staggered Quadrature Phase Shift Keying (SQPSK), Minimum Shift Keying (MSK), 8-ary Phase Shift Keying (8-PSK), and 16-ary Quadrature Amplitude Modulation (16-QAM). Each of these systems generated the aforementioned modulation types with signal-to-noise ratios of 25db, 10db, 5db, and 1db. Each signal generated contained 1024 points with a sampling frequency of 16.0 Hz. The noise generator utilizes a noise bandwidth of 1.0 Hz. The next few pages show the design of each of the systems as they were created in ComDisco's SPW. Figure 6 illustrates a system that utilizes two complex random number generators to produce inphase and quadrature data required as input to the SQPSK modulator. This signal is then split into five different signals, each with a different signal to noise ratio (SNR) of 25db, 10db, 5db, 1db, and a clean signal (no noise). The noise that is introduced into the signal is Gaussian white noise. Figures 6, 8, 10, and 12 have the same general construction except that a different modulation block (SQPSK, 8-PSK, 16-QAM, and MSK) is used for each model. The signals generated by each of the systems in Figures 6, 8, 10, and 12, are displayed using the signal display editor in SPW and are illustrated in Figures 7, 9, 11, and 13. In Figure 7, the first signal displayed is an SQPSK modulated signal with a SNR of 1db. It is impossible to distinguish SQPSK pulses from this figure by looking at it. The second signal has a SNR of 5db and it is still

nearly impossible to distinguish signal pulses. The third signal has a SNR of 10db and it is possible, although difficult, to distinguish pulses from the noise. The fourth signal has recognizable pulses with a SNR of 25db and the last signal is a pure signal with no noise at all. Figures 7, 9, 11, and 13 are all displayed in the same format as described above. Figure 6 is the SQPSK system model that generates the signals displayed in Figure 7. Figure 8 is the 8-PSK system model that generates the signals displayed in Figure 9. Figure 10 is the 16-QAM system model that generates the signals displayed in Figure 11. Figure 12 is the MSK system model that generates the signals displayed in Figure 13.

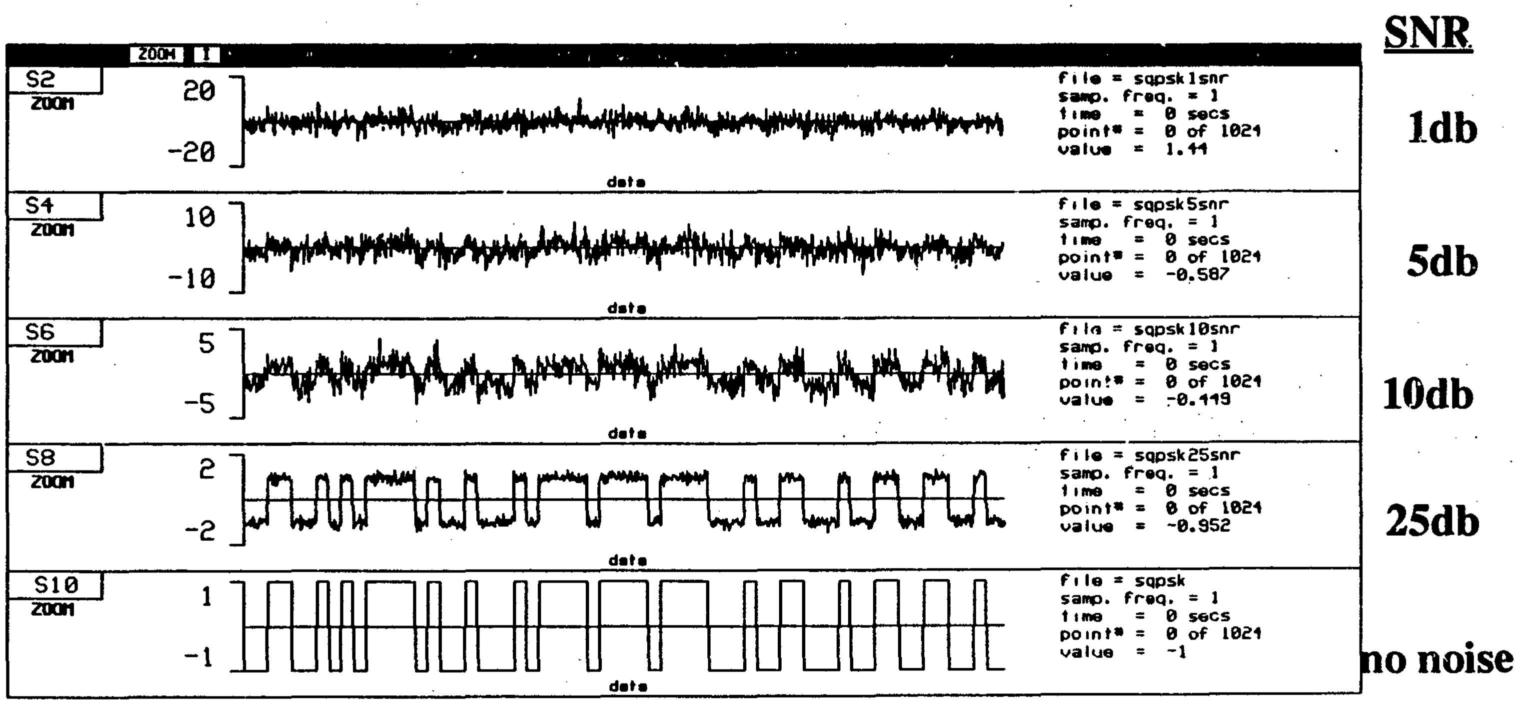
## 5.3 Simulation Description:

The NWRS classifies four different waveforms (no noise added) using the Kohonen's Self-Organizing Feature Map paradigm. The network is trained by repeatedly presenting the waveform signal epochs to the input layer of the NWRS. When presentation of the four waveforms no longer causes the interconnection weights to change, the network is completely trained. Once training is complete, weight adaptation is discontinued and the system is tested to see if it can correctly identify each of the distinct waveforms. Correct recognition of the waveform is obtained if the best matching exemplar (in the output layer) is the exemplar where the original pattern was stored. The best matching exemplar is the exemplar that is the shortest Euclidean distance from the signal epoch input vector (as shown in Figure 14).

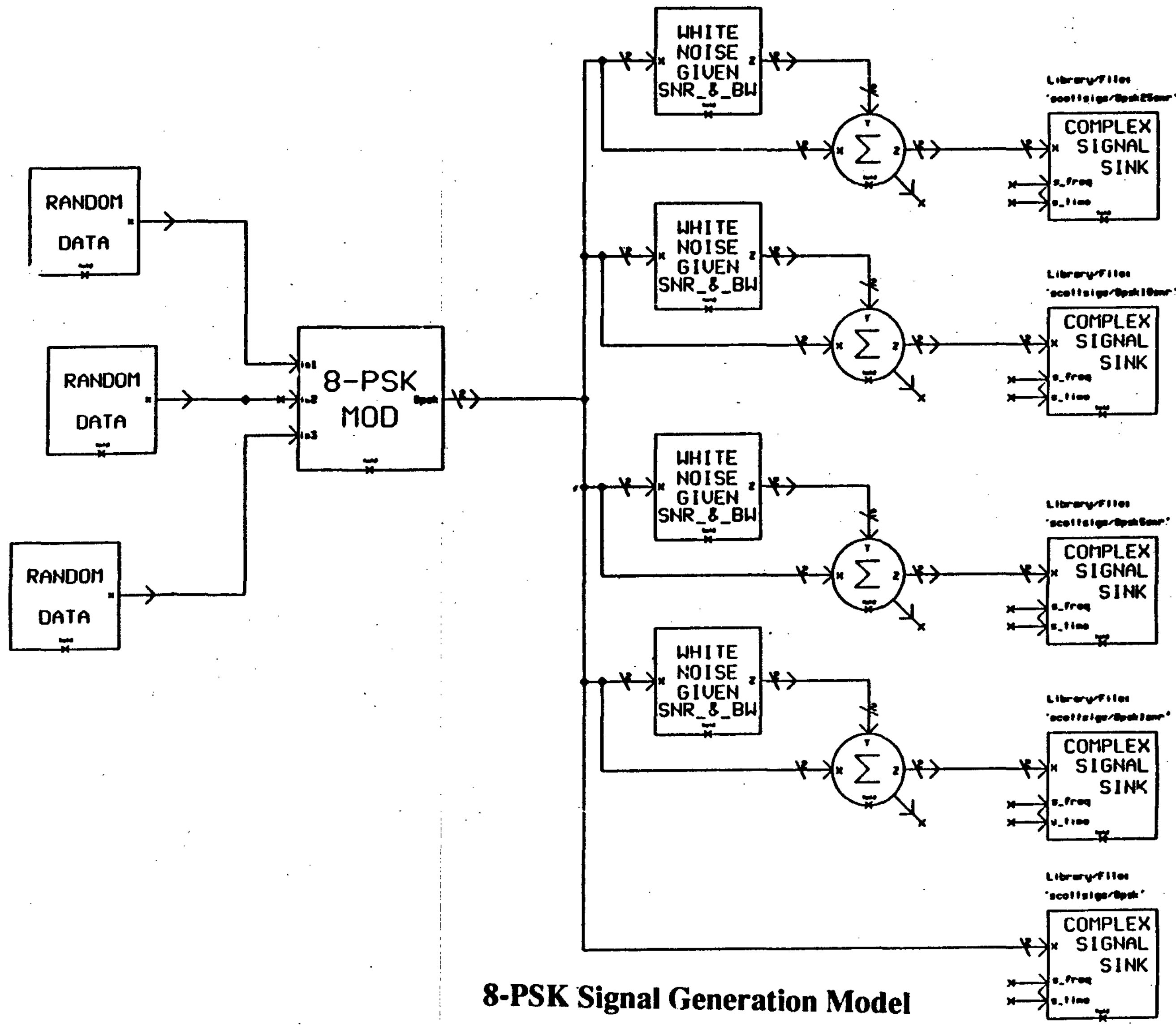
The network is further tested by presenting the input layer with waveform signal epochs that were generated with signal-to-noise ratios of 25db, 10db, 5db, and 1db. The purpose of this type of testing is to determine if the NWRS can correctly classify waveforms that have been corrupted by Gaussian white noise. The NWRS showed excellent results in the ability to classify waveforms that had signal-to-noise ratios (SNR) of 25db. The network correctly classified these signals 100% of the time. When the SNR was reduced to 10db, the network was again able to



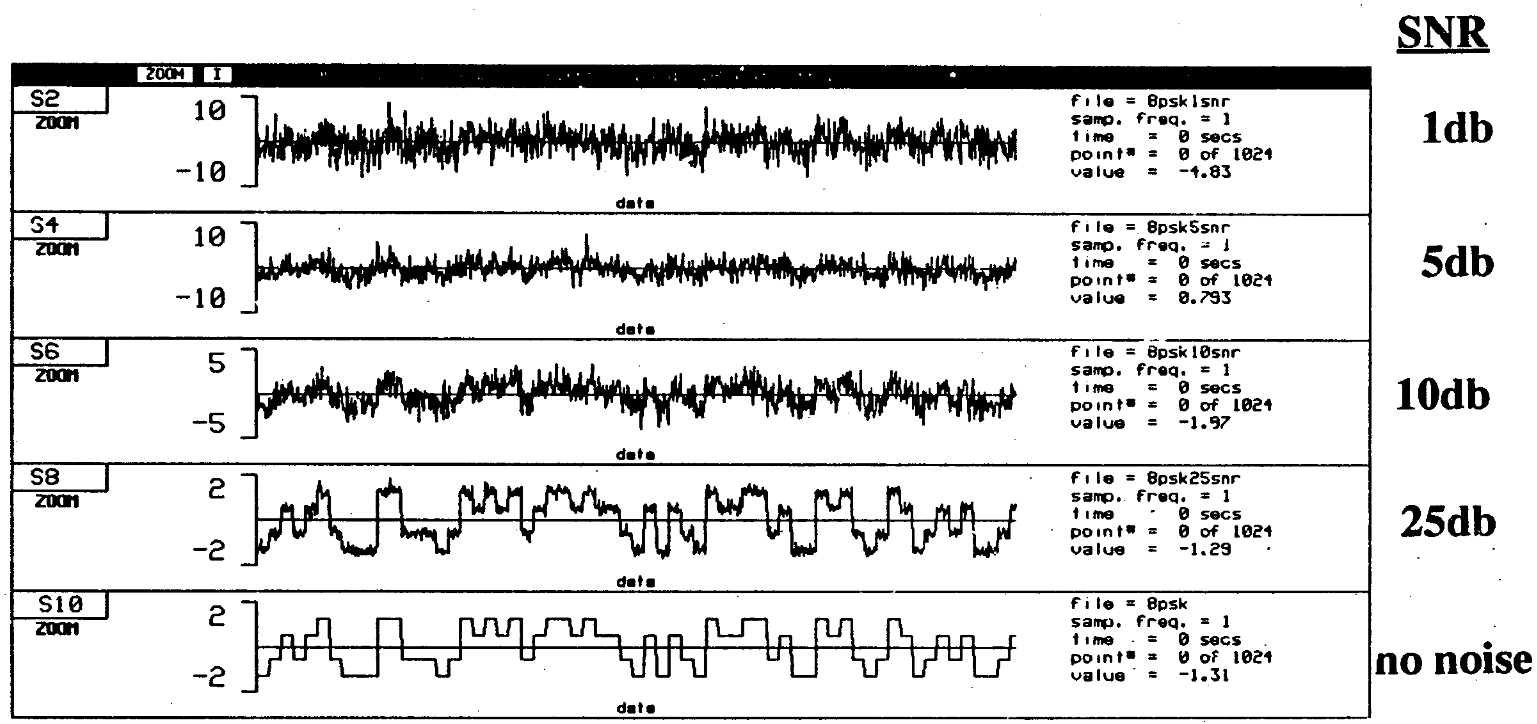

**SQPSK Signal Generation Model**
**Figure 6**



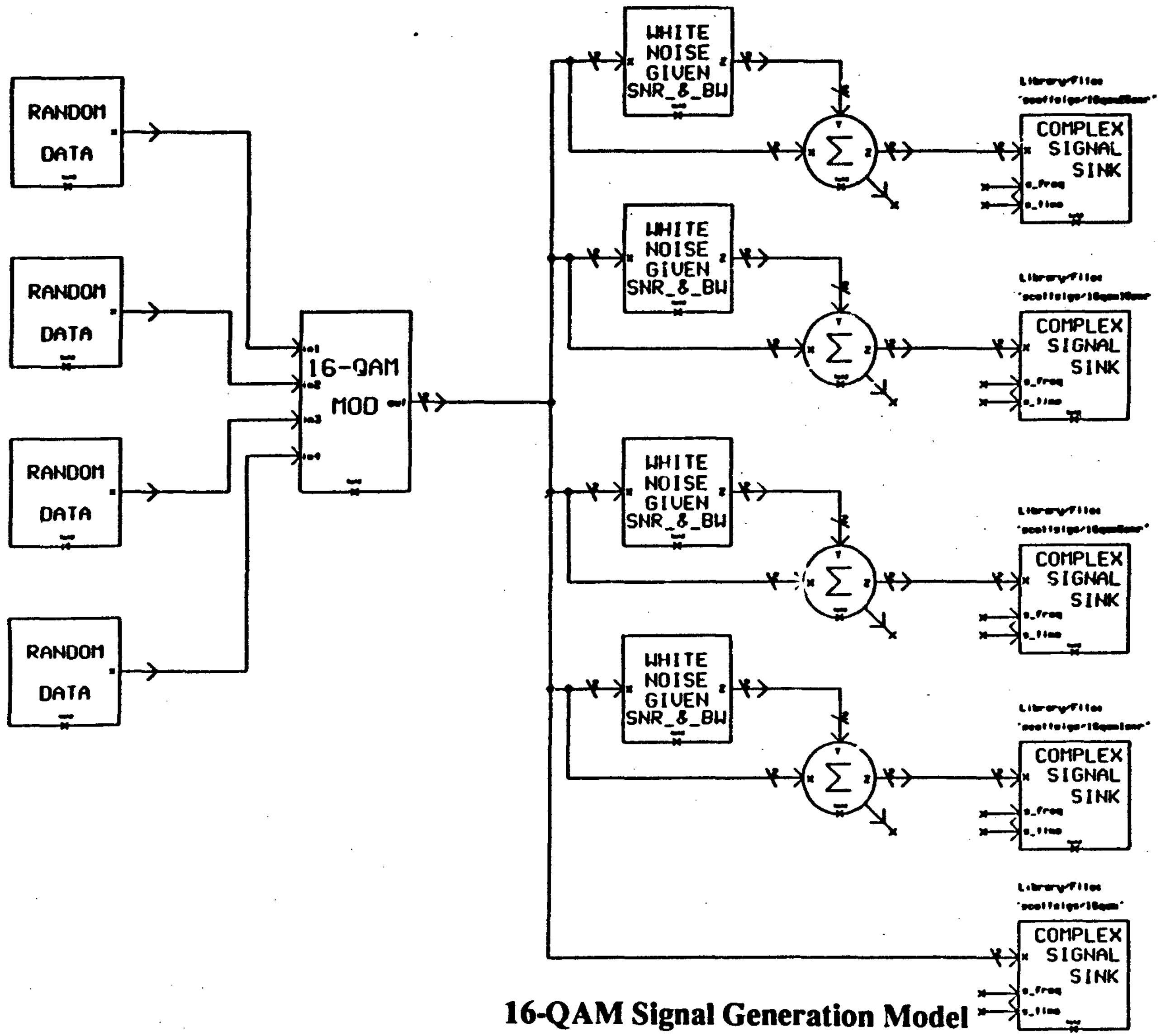

**SQPSK Signal Graph**
**Figure 7**



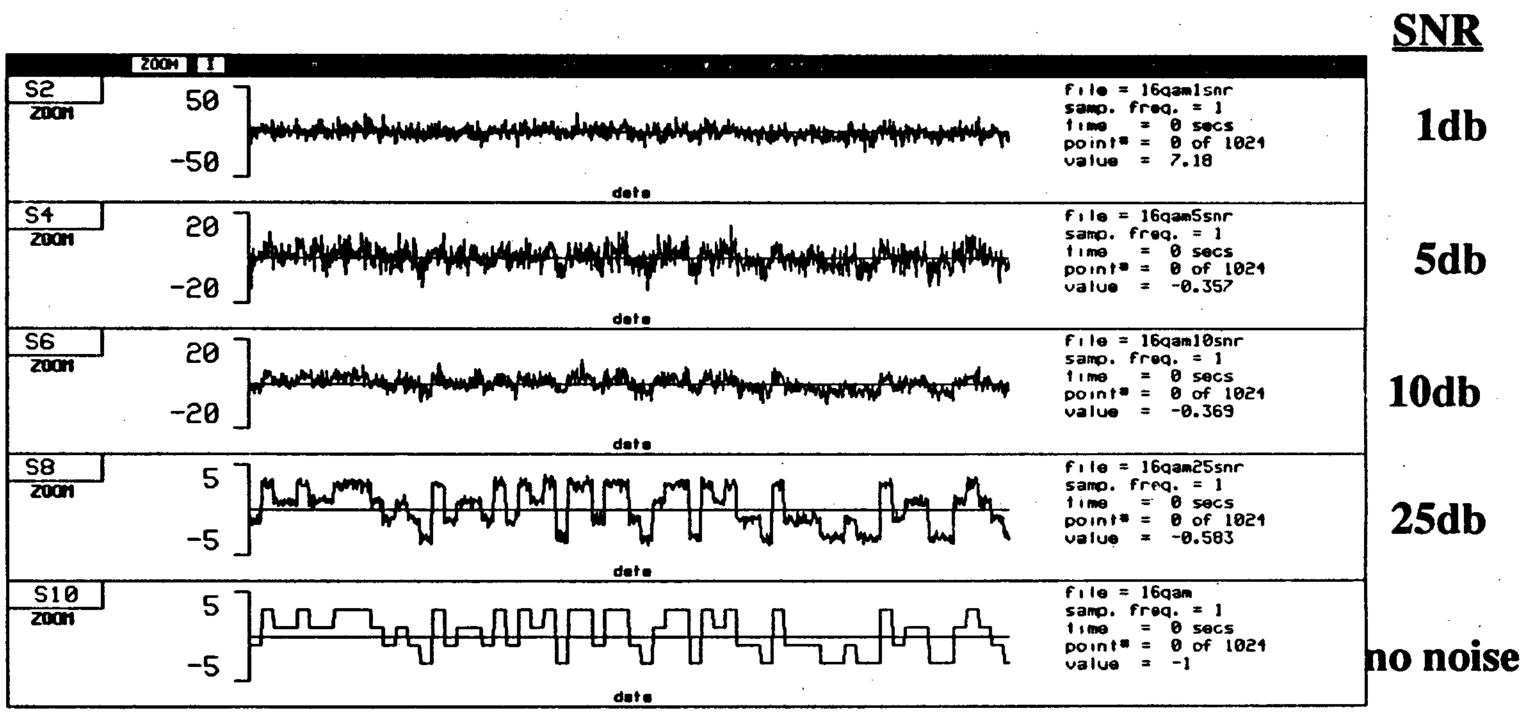

**8-PSK Signal Generation Model**
**Figure 8**




**8-PSK Signal Graph**
**Figure 9**

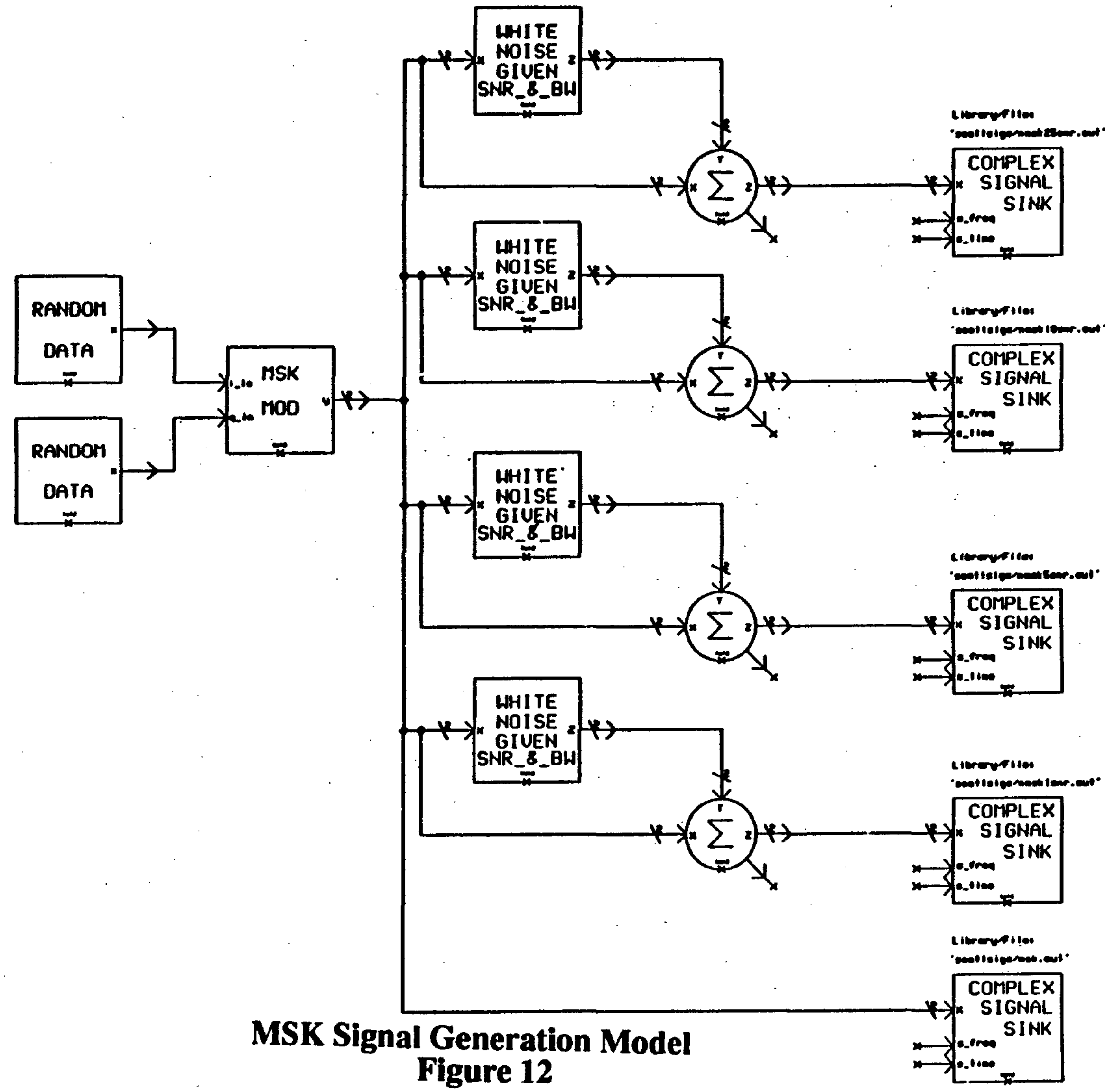



**16-QAM Signal Generation Model**
**Figure 10**

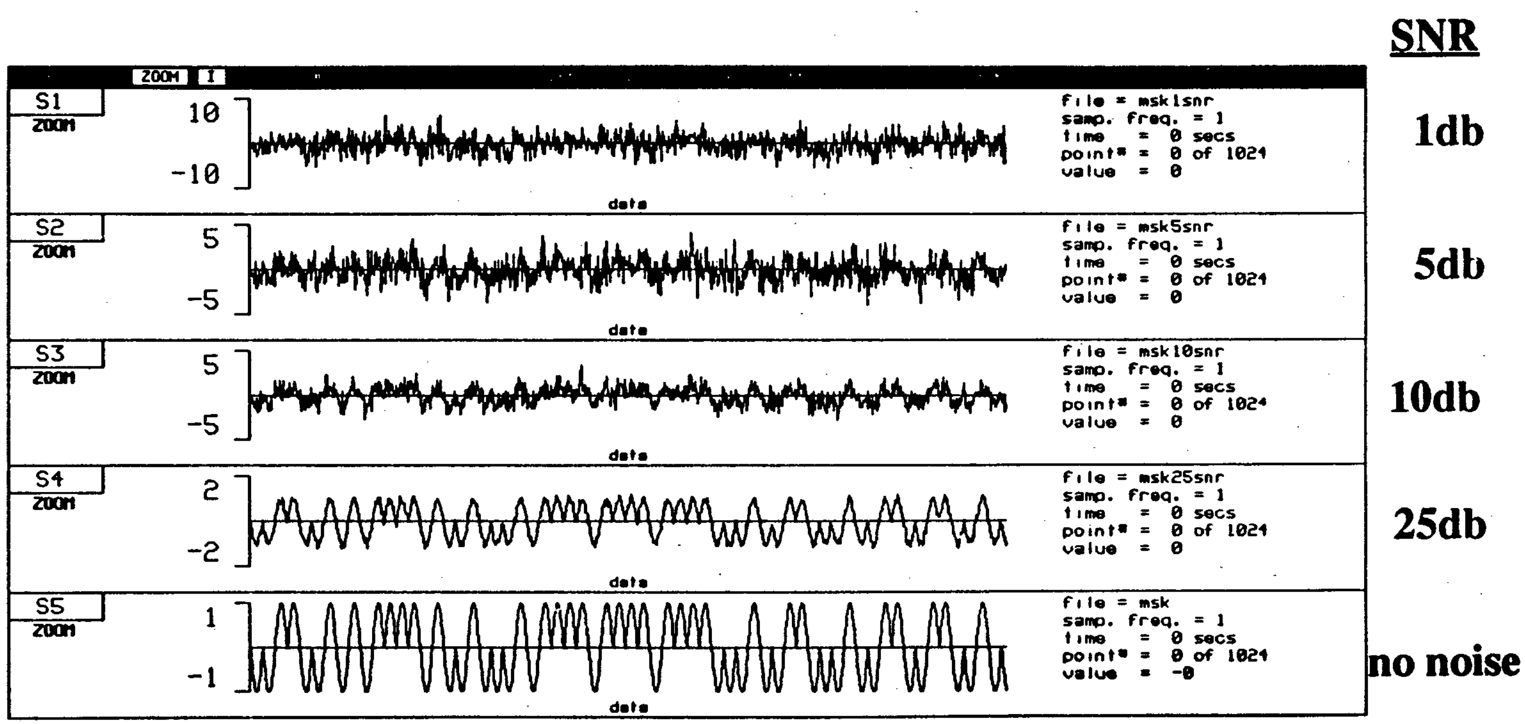



**16-QAM Signal Graph**

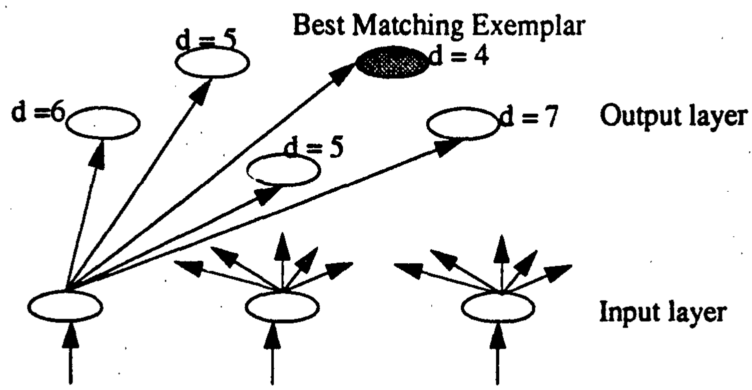
**Figure 11**




**MSK Signal Generation Model**
**Figure 12**




MSK Signal Graph
Figure 13




Waveform Signal Epoch Input Vector

Classification of Signal Epoch

Fig 14

correctly classify the waveform signal epoch by matching the waveform to an exemplar that was close to the exemplar originally designated as that particular waveforms exemplar. Once the SNR dropped to 5db, the network had great difficulty classifying the SQPSK, 16-QAM, and the 8-PSK signal epochs. It was able to classify the MSK waveform probably due to the great difference in shape between the MSK modulation and the other three modulation types. The NWRS could not classify any of the signal epochs with a signal-to-noise ratio of 1db. The following tables show some of the results of testing the NWRS with various signal epochs. The numbers in the table represent which exemplar the signal epoch best matched to. There are 1024 possible exemplars to match to. Exemplar 1024 is a garbage node and when a signal matches to it, this means that the network doesn't know what type of signal was just presented.

**Table 1: NWRS 10 Training Iterations**

| MODULATION | Pure Signal | 25db SNR | 10db SNR | 5db SNR | 1db SNR |
|---|---|---|---|---|---|
| SQPSK | 530 | 549 | 530 | 1024 | 1024 |
| 16-QAM | 1018 | 1015 | 1024 | 1024 | 1024 |
| 8-PSK | 9 | 6 | 5 | 1024 | 1024 |
| MSK | 403 | 415 | 416 | 425 | 1024 |

**Table 2: NWRS 20 Training Iterations**

| MODULATION | Pure Signal | 25db SNR | 10db SNR | 5db SNR | 1db SNR |
|---|---|---|---|---|---|
| SQPSK | 532 | 540 | 559 | 1024 | 1024 |
| 16-QAM | 1022 | 1019 | 1024 | 1024 | 1024 |
| 8-PSK | 9 | 6 | 5 | 1024 | 1024 |
| MSK | 311 | 309 | 317 | 299 | 1024 |

**Table 3: NWRS 30 Training Iterations**

| MODULATION | Pure Signal | 25db SNR | 10db SNR | 5db SNR | 1db SNR |
|---|---|---|---|---|---|
| SQPSK | 529 | 513 | 594 | 1024 | 1024 |
| 16-QAM | 1018 | 1021 | 1024 | 1024 | 1024 |
| 8-PSK | 9 | 5 | 13 | 1024 | 1024 |
| MSK | 319 | 297 | 306 | 257 | 1024 |

The increase in training iterations did not significantly effect the performance of the NWRS. An attempt was made to increase the training iterations by several orders of magnitude. As the following data shows, there was no significant improvement or degradation noticed. The exemplars converged to the exemplars listed in the following example after 50 iterations. No change was noticed from iteration 51 through to iteration 1000.

**Table 4: NWRS 1000 Training Iterations**

| MODULATION | Pure Signal | 25db SNR | 10db SNR | 5db SNR | 1db SNR |
|---|---|---|---|---|---|
| SQPSK | 534 | 513 | 594 | 1024 | 1024 |
| 16-QAM | 1019 | 1015 | 1024 | 1024 | 1024 |
| 8-PSK | 9 | 15 | 33 | 1024 | 1024 |
| MSK | 302 | 302 | 294 | 290 | 1024 |

An attempt was made to train the network using the corrupted data from the 25db SNR signal epochs instead of training the NWRS of clean data. The thought being that perhaps since the performance was based on the ability of the NWRS to classify corrupted signal epochs, train-

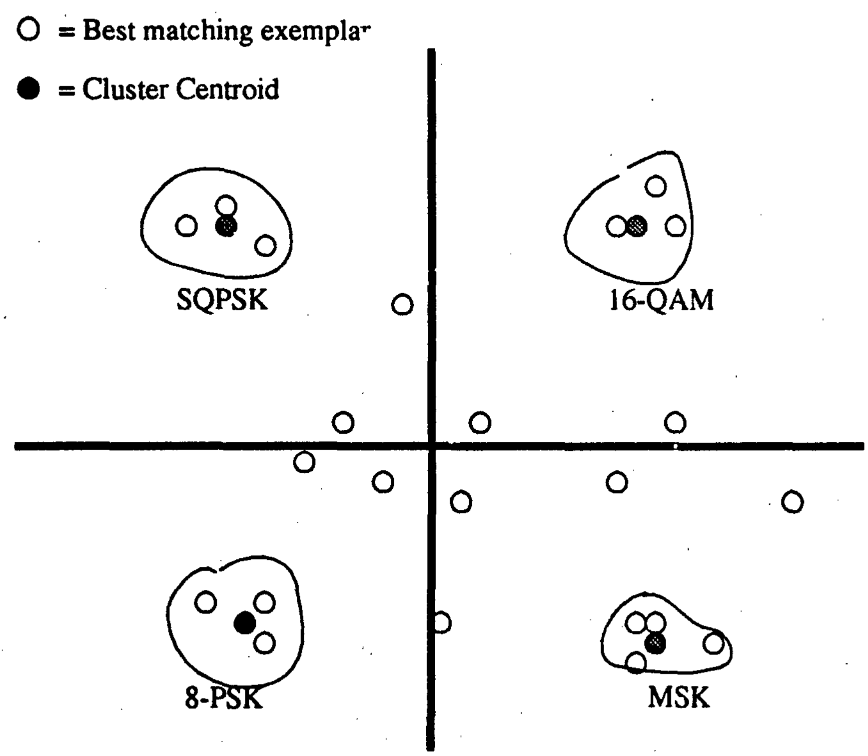
ing it on slightly noisy signal epochs would be a more representative training set. As the following data shows, there was no significant improvement or degradation noticed.

**Table 5: NWRS 30 Training Iterations with 25db SNR data.**

| MODULATION | Pure Signal | 25db SNR | 10db SNR | 5db SNR | 1db SNR |
|---|---|---|---|---|---|
| SQPSK | 568 | 513 | 568 | 1024 | 1024 |
| 16-QAM | 1003 | 1021 | 1024 | 1024 | 1024 |
| 8-PSK | 1 | 17 | 21 | 1024 | 1024 |
| MSK | 305 | 305 | 305 | 260 | 1024 |

The NWRS is based on the Kohonen Self-Organizing Feature Map which utilizes the K-means clustering algorithm. This algorithm uses the Euclidean distance to determine what cluster of output vectors a given input vector belongs too. When using Gaussian white noise, the mean of the noise over the entire waveform is zero. Therefore, as far as the overall distance from this noisy input vector to its correct cluster is concerned, there is almost no difference between a vector with 25dbSNR using Gaussian noise (overall change being nearly zero from original vector) and one without any Gaussian noise. As the SNR decreased, the power of the noise increased, greatly disturbing the overall mean of the signal epoch. The difference between the mean of the original signal epoch and the mean of the noisy signal epoch differed by such a wide margin that the NWRS could not correctly classify the noisy signal epoch as a member of the original signal epochs cluster. Figure 15 illustrates graphically how the NWRS classifies an input pattern to be a member of a learned signal cluster or just an unrecognizable noisy signal. During training, the NWRS self-organizes its memory to reflect the number of different signal epochs it was trained on. Each of these training patterns become the central point (centroid) of the memory map. The area around the centroid is called the body of the cluster. If a best matching exemplar falls within the body of a defined cluster, the signal can be classified as a member of that cluster and it is now known what type of signal epoch was just presented to the NWRS.

The exemplars that did not fall into a cluster were not recognized by the NWRS. These exemplars were not recognized mostly because the SNR was to low. The power of the noise greatly overshadowed the characteristics of the actual signal. If some type of preprocessor were




Signal Clustering for Identification
Figure 15

introduced to normalize the noisy signal epoch to the power level of the original signal epoch, the classification results of the NWRS would be greatly improved. Other types of noise and jamming will tend to have significant effects on the NWRS's ability to correctly classify the waveform depending on the effect these outside interferers have on the mean of the signal epoch and the amount of power added to the signal. More work needs to be performed in this area. Once the NWRS correctly classifies the waveform, the receiver can utilize the proper techniques to decipher the transmitted message.

## 6.0 Results and Recommendations:

Kohonen's Self-Organizing Feature Map has shown positive preliminary results in the area of waveform recognition given a noisy environment. The simulation was created as a toy problem to illustrate some of the major characteristics of the Kohonen algorithm. Several variables can be modified to explore the performance of the algorithm. The most prevalent modification would be to adjust the nearest neighbor excitability. In the current simulation, the

neighborhood is decreased after every three iterations until the neighborhood becomes one. By increasing the number of training iterations and adjusting the neighborhood every fourth or every sixth iteration, the resulting memory map might be better defined and therefore be more noise tolerant. As the neighborhood decreases in size, the amount each weight is adjusted also is decreased. This weight adjustment is another factor that can be altered for experimentation.

Future development in this application area will include expanding this simulation to run in ComDisco's "Signal Processing Workstation" (SPW) communications simulation environment. This will allow the use of many different modulation types for classification. It will be interesting to see how the classification of the waveform performs if noise jammers are introduced in addition to Gaussian white noise. By implementing NWRS as a single "block" in the SPW environment, it can be incorporated into an entire system to study the effects of the NWRS when used in conjunction with conventional noise cancellation filters. Utilizing the SPW environment will also provide a full test suite of analysis tools that can be used to tune the NWRS for better performance. It is noped that the NWRS in combination with SPW will provide a test environment that may help us understand some of the overall complexities of introducing neural networks into the world of communications.

# Bibliography


Advances in Neural Information Processing, Vol I. "Neural Net Receivers in Multiple-Access Communications", Morgan Kaufman Publishing, 1989, pg 272-280.

Advances in Neural Information Processing, Vol II. "A Neural Network for Real-time Signal Processing", Morgan Kaufman Publishing, 1989, pg 248-255.

DARPA Neural Network Study, October 1987 - February 1988. Fairfax, Virginia: AFCEA International Press.

Kosko, Bart (1992), Neural Networks and Fuzzy Systems. Englewood Cliffs, New Jersey: Prentice-Hall Inc.

Lippmann, Richard P., An Introduction to Computing with Neural Networks. IEEE ASSP Magazine, April 1987.

Skylar, Bernard (1988), Digital Communications - Fundamentals and Applications. Englewood Cliffs, New Jersey: Prentice-Hall, Inc.

Signal Analytics Corporation (May 1991), Communications Signal Recognition and Demodulation Via Neural Networks, RL-TR-91-51.

Shyne, Scott (1990), Master's Thesis: "The Implementation of Neural Networks as Adaptive Pattern Classifiers". Ann Arbor, Michigan: University Microfilm International.



*U.S. GOVERNMENT PRINTING OFFICE: 1993-710-093-60136

**MISSION**

**OF**

**ROME LABORATORY**

*Rome Laboratory plans and executes an interdisciplinary program in research, development, test, and technology transition in support of Air Force Command, Control, Communications and Intelligence ($C^3I$) activities for all Air Force platforms. It also executes selected acquisition programs in several areas of expertise. Technical and engineering support within areas of competence is provided to ESD Program Offices (POs) and other ESD elements to perform effective acquisition of $C^3I$ systems. In addition, Rome Laboratory's technology supports other AFSC Product Divisions, the Air Force user community, and other DOD and non-DOD agencies. Rome Laboratory maintains technical competence and research programs in areas including, but not limited to, communications, command and control, battle management, intelligence information processing, computational sciences and software producibility, wide area surveillance/sensors, signal processing, solid state sciences, photonics, electromagnetic technology, superconductivity, and electronic reliability/maintainability and testability.*

END

FILMED

DATE: 4-93

DTIC